

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
26 April 2001 (26.04.2001)

PCT

(10) International Publication Number
WO 01/29001 A2

(51) International Patent Classification⁷: C07D 209/00

(21) International Application Number: PCT/US00/28382

(22) International Filing Date: 13 October 2000 (13.10.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/159,777 15 October 1999 (15.10.1999) US

(71) Applicant (for all designated States except US): NAME-TREE, INC. [US/US]; 3959 Pender Drive, Suite 201, Fairfax, VA 22030 (US).

(72) Inventor; and

(75) Inventor/Applicant (for US only): LEONARD, Kenneth, F. [US/US]; 26 Orange Court, Hillsborough, CA 94010 (US).

(74) Agents: GREENBERG, Robert, A. et al.; Foley, Hoag & Eliot, LLP, One Post Office Square, Boston, MA 02109 (US).

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

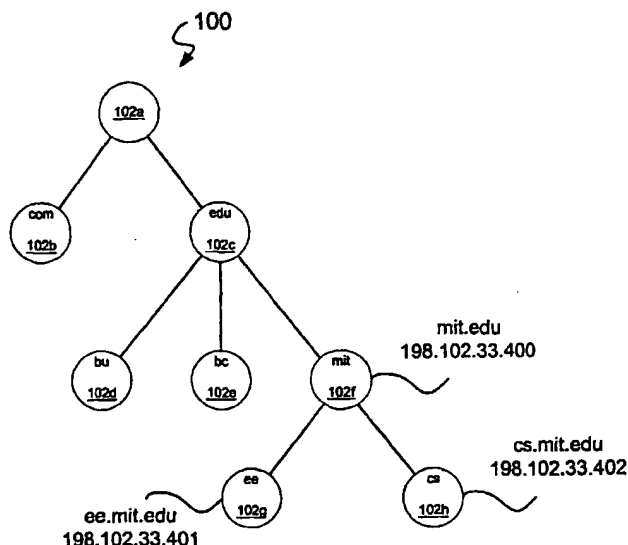
(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— Without international search report and to be republished upon receipt of that report.

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: DOMAIN NAME SEARCHING



(57) Abstract: A method of handling a network domain name query includes receiving a query for a network domain name from a remote network computer where the domain name query is capable of matching multiple, different domain names, and identifying one or more network domain names matching the received network domain name.

BEST AVAILABLE COPY

WO 01/29001 A2

DOMAIN NAME SEARCHING

5 Reference to Related Applications

This application claims priority for subject matter disclosed in pending U.S. Provisional Application Serial No. 60/159,777, entitled "SYSTEMS AND METHODS FOR ALLOWING EXTENDED DOMAIN NAME SPACES", which is incorporated by
10 reference in its entirety herein.

This application relates to U.S. Application Serial No. _____, entitled "DERIVATIVE DOMAIN NAMES", filed on the same day as the present application.

15 Background

Much like homes on a street, computers and other network devices send information to each other using addresses. For example, a 32-bit Internet Protocol (IP) address can identify computers on a TCP/IP (Transmission Control Protocol/
20 Internet Protocol) network. For example, a computer on the network may have an IP address of "198.102.33.402". Network routers, analogous to mailmen, use the IP address to forward messages to their destination.

Because IP addresses are somewhat cryptic and,
25 potentially, subject to change, programs and users rarely refer to network devices by their binary addresses, but can instead identify network devices using domain names. For example, instead of specifying "198.102.33.402", a program or user can identify a device with the more memorable text of
30 "cs.mit.edu". Since networks ultimately deal with binary addresses, programs known as "name servers" translate domain name text into the corresponding network addresses. The translation process is known as "name resolution".

FIG. 1 illustrates a domain name system (DNS) used by the Internet, FTP (File Transfer Protocol), and other network applications. DNS uses a tree 100 to represent a hierarchy of domains. Nodes 102 in the tree 100 have domain labels.

5 For example, node 102h has a label of "cs". The full domain name of a node 102 in the tree 100 is a sequence of labels, separated by periods, that define a path from node to the tree 100 root 102a. For example, the full domain name of node 102h is "cs.mit.edu". As shown, a node in the tree 100

10 can correspond to a host address (e.g., the address of a computer, network device, or sub-network). For example, as shown, the domain "cs.mit.edu" corresponds to IP address "198.102.33.402".

As described above, a name server handles the task of

15 translating a domain name into a network address. As shown in FIG. 2, a name server can handle domain name queries for more than one node. For example, a name server at node 102f can maintain a database 104 of domain names and their corresponding network addresses for nodes 102f-102h composing

20 zone 106. The name server can, thus, offer name resolution for domain names "mit.edu", "ee.mit.edu", and "cs.mit.edu". The server at node 102f is, thus, known as an "authoritative server" for the zone.

To reduce the burden of name resolution on any one name

25 server, DNS distributes domain name information across different name servers. As a result, each name server may only be able to resolve a limited subset of domain names in the DNS tree 100. As such, if a name server cannot resolve a domain name query, the name server can ask another name

30 server (e.g., a name server higher in the tree 100) to resolve the domain name. As shown in FIG. 3, several computers may hand off a domain name query before successful resolution.

In greater detail, FIG. 3 illustrates a network computer 108 running a network application such as a web-browser. After a user enters a URL (Universal Resource Locator) that includes a domain name, an entity known as a resolver queries
5 a name server for translation of the domain name to an IP address. As shown, a query for "ee.mit.edu" may pass between nodes 102d and 102c, before reaching a node 102f having a name server able to resolve the domain name. A message including the IP address, "198.102.33.401", corresponding to
10 the "ee.mit.edu" domain name query retraces a path back to the computer 108 that originally issued the domain name query. Using the received IP address, the computer may begin communicating with the host, for example, to download web-pages or other information.

15 To speed name resolution, name servers may cache domain name/IP address pairs as discovered. For example, after successful resolution of the query shown in FIG. 3, name servers at nodes 102d and 102c may update their databases to reflect that "198.102.33.401" corresponds to "ee.mit.edu".
20 These nodes 102c, 120d may also cache information indicating that the name server at node 102f ultimately resolved the query. Thus, in the future, node 102d can bypass node 102c, and directly query node 102f for "ee.mit.edu" resolution.

25

Summary

In general, in one aspect, the invention features a method of handling a network domain name query. The method includes receiving a network domain name query capable of
30 matching multiple, different domain names from a remote network computer, and identifying one or more network domain names matching the received network domain name.

Embodiments may include one or more of the following features. The domain name query may include one or more

operator. The operator can include characters such as a "*",
"?", "~", "!", or "#". The wildcard may include an absence
of characters in a period delimited sequence of domains. The
wildcard may stand for an entire domain label or a portion of
5 an entire domain label. The operator may include a Boolean
operator.

The method may further include transmitting the one or
more identified domain names to the remote network computer,
for example, by constructing user interface instructions
10 (e.g., HTML) including the identified domain names.
Transmitting the identified domain names may include
transmitting a network address of a host storing the
constructed user interface instructions.

The method may further include transmitting a network
15 address corresponding to at least one of the identified
domain names to the remote computer. The method may further
include determining if identifying yields only one matching
domain name and transmitting a network address corresponding
to the one matching domain name.

20 Receiving a query for a network domain name may include
receiving a DNS (domain name service) query issued by a
resolver.

The method may also include providing user interface
instructions for transmission to a remote computer and
25 presentation to a user and receiving a query generated in
response to user interaction with the user interface
instructions.

In general, in another aspect, the invention features a
computer program product, disposed on a computer readable
30 medium, for handling a network domain name query. The
computer program product includes instructions for causing a
processor to receive a network domain name query capable of
matching multiple, different domain names from a remote

network computer, and identify one or more network domain names matching the received network domain name.

Advantages will become apparent in view of the following description, including the figures and the claims.

5

Brief Description of the Drawings

FIG. 1 is a diagram of a domain name space.

FIG. 2 is a diagram including a domain name zone.

FIG. 3 is a flow-diagram illustrating domain name
10 translation.

FIG. 4 is a flow-diagram illustrating pattern matching of a domain name query that includes a wildcard.

FIG. 5 is a screenshot of a web-page display of matches for a domain name query.

15 FIG. 6 is a screenshot of domain name search engine.

FIG. 7 is a flow-chart of process for identifying matches for a domain name query.

20

Detailed Description

FIG. 4 illustrates operation of a name server 110 that can resolve domain name queries capable of matching more than one different domain name. For example, as shown in FIG. 4, name server 110 receives a domain name query 112 for
25 resolution of "*.mit.edu". In this example, the asterisk, "*", 116 represents a wildcard operator that can match an number of consecutive characters (e.g., ASCII or Unicode characters). Thus, when searching the domain name database 104 for domain names matching the query 112, the name server
30 110 interprets the wildcard operator to identify domain names that match the following pattern: <any number of characters> followed by <.mit.edu>. As shown, the name server 110 identifies "ee.mit.edu" 114a and "cs.mit.edu" 115b as matches for the "*.mit.edu" query. The ability to resolve domain

names capable of matching different domain names can enable users to quickly access a particular domain even though they may have only a sketchy memory of the domain name.

As shown in FIG. 4, the wildcard operator is an asterisk, however, other characters may be used, such as a question mark, "?", exclamation point, "!", and so forth. Different wildcard operators may express different pattern matching instructions. For example, the asterisk, "*", may instruct the name server to match the "*" with any number of characters, while a number sign, "#", may instruct the name server to match the "#" with only a single character. For example, "ee.mit.edu" may match "*.mit.edu", but not "#.mit.edu" since the "ee" portion of "ee.mit.edu" is more than the one character specified by the number sign, "#", character.

A domain name query may include more than one wildcard operator. For example, "ee.mit.edu" may match a query of "*#.mit.com". Additionally, the wildcards may match characters crossing over more than one domain label and can include delimiting periods. For example, "ai.cs.mit.edu" may match "*.mit.edu". Additionally, wildcard operators may be intermixed with non-wildcard elements (i.e., elements requiring an exact character for character match) within domain name labels. For example, "ee.mit.edu" matches "#e.mit.edu" or "e*.mit.edu".

In addition to wildcard characters, a wildcard operator may be indicated by the absence of characters. For example, absence of non-blank characters before or after a domain name delimiter (e.g., a period) can indicate a wildcard operator. For example, the domain name query "ee..edu" or "ee. .edu" may instruct the name server to match any domain names starting with "ee." and ending with ".edu" since no non-blank characters occur between the delimiting periods, ".".

In addition to, or in lieu of, wildcard operators, the domain name query may include expressions that include operators. For example, the domain name query may include an expression including boolean operators. For example, a
5 domain name query may take the form `("ee" or "cs").mit.edu`. In this case the expression, `"ee" or "cs"`, includes the Boolean OR operator. Evaluation of the expression by the name server results in matches for both `"ee.mit.edu"` and `"cs.mit.edu"`.

10 A wide variety of other operators may be used. For example, operators may express relationships between different text components of a domain name. For example, a domain name query of `("ee" /within=2characters "program").mit.edu` would match `"eeprogram.mit.edu"`, but not
15 `"ee_grad_program.mit.edu"` since in the latter example more than two characters separate `"ee"` and `"program"`.

A wide variety of pattern matching algorithms may be used to match domain name queries with domain name records. Additionally, many other wildcard and other operators may be
20 used. Further, the pattern matching need not be limited to characters of a particular language or character set. That is, `"*Fraternity.Boston.com"` may match `"αβδ-Fraternity.Boston.com"`.

In addition to handling wildcards, the name server may
25 provide a database that stores "derivative" domain names such as domain names having spaces in domain labels, such as `"Movies at .boston.com"`. Additionally, domain names stored in the database may include characters from different character sets. For example, an domain name of `"αβδ`
30 `Fraternity at Boston.com"` can include characters from a Greek character set in addition to characters from an ASCII character set.

In some circumstances, the location of the operator within the query may be restricted. For example, for domain

name queries received by an authoritative server from a resolver, the query must have sufficient information for the DNS system to route the query to the authoritative server. For example, if a name server is the authoritative server for a "boston.com" zone, operators appearing in the ".boston.com" portion of the query may prevent the query from reaching the zone's authoritative server. That is, "* at.boston.com" will reach the "boston.com" authoritative name server, but "#oston.com" may not.

10 As shown in FIG. 5, after identifying domain name matches for a query, the name server (or other program) may dynamically construct user interface instructions (e.g., HTML (HyperText Markup Language) instructions) that include the identified domain names. The name server can return the IP address of a host storing the constructed user interface instructions. A subsequent message (e.g., an HTTP (HyperText Transfer Protocol) message) to the host can prompt delivery of the instructions for display to the user.

For example, as shown in FIG. 5, a user has entered a URL (Universal Resource Locator) into a browser address field 120. The URL includes a domain name having an asterisk for a wildcard operator. After a resolver submits a query for the URL on behalf of the browser, a name server, in accordance with the techniques described herein, can identify multiple, different matches for the query and dynamically construct a web-page listing the matches. The name server can then send the browser's resolver the IP address for a host storing the constructed web-page. As shown, in addition to matching entries, the name server has included a brief description of the content offered by the matches. Since the returned IP address may not correspond to an IP address satisfying the query, the name server may set to the "Time to Live" field to 0, thereby preventing other name servers from caching the IP address.

Instead of the user transparent operation of FIG. 5 where the resolver and name server operate "behind-the-scenes", FIG. 6 shows a domain name search engine. For example, FIG. 6 may be an HTML form 128 that includes a user input control such as a text entry field 129. As shown, after the user submits a domain name query into the form, the search engine transmits an HTML page including matching domain names to the users computer for display.

FIG. 7 illustrates a process 130 for identifying matches to a domain name query. The process 130 includes receiving 132 a domain name query capable of matching different domain names. For example, the process 130 may receive a DNS query from a resolver (e.g., see FIG. 5) or from a search engine form (e.g., see FIG. 6). The process 130 then searches 134 stored domain names to identify entries matching the query. For example, the process 130 may use a pattern matching algorithm to process wildcard operators and/or evaluate Boolean expressions to identify matches.

If the process 130 cannot identify any matches (not shown), the process 130 may return an indication of failure. Alternatively, the process 130 may query a different name server for resolution.

If the process 130 identifies one or more matches, the process 130 can present 138 the matches, for example, by dynamically constructing an HTML web-page. The term "presenting", as used herein, indicates the process 130 can cause presentation of the matches to a user (e.g., visually or audibly) or can present the matches to a program as data.

In some embodiments, the process 130 can determine whether only one domain name matches 136 the domain name query. In such cases, since no ambiguity exists, the process 130 can return 140 the corresponding network address.

The techniques described herein are not limited to any particular hardware or software configuration; they may find

applicability in any computing or processing environment.
The techniques may be implemented in hardware or software, or
a combination of the two. Preferably, the techniques are
implemented in computer programs executing on programmable
5 computers that each include a processor, a storage medium
readable by the processor (including volatile and non-
volatile memory and/or storage elements), at least one input
device, and one or more output devices.

Each program is preferably implemented in high level
10 procedural or object oriented programming language to
communicate with a computer system. However, the programs
can be implemented in assembly or machine language, if
desired. In any case the language may be compiled or
interpreted language.

15 Each such computer program is preferably stored on a
storage medium or device (e.g., CD-ROM, hard disk, or
magnetic disk) that is readable by a general or special
purpose programmable computer for configuring and operating
the computer when the storage medium or device is read by the
20 computer to perform the procedures described herein. The
system may also be considered to be implemented as a
computer-readable storage medium, configured with a computer
program, where the storage medium so configured causes a
computer processor to operate in a specific and predefined
25 manner.

Other embodiments are within the scope of the following
claims.

What is claimed is:

1. A method of handling a network domain name query,
the method comprising:

receiving a query for a network domain name from a
remote network computer, the domain name query capable of
5 matching multiple, different domain names; and
identifying one or more network domain names matching
the received network domain name.

2. The method of claim 1, wherein the domain name query
10 comprises at least one operator.

3. The method of claim 2, wherein the at least one
operator comprises at least one wildcard.

15 4. The method of claim 3, wherein the at least one
wildcard comprises more than one wildcard.

5. The method of claim 3, wherein the at least one
wildcard comprises at least one of the following characters:
20 "*", "?", "~", "!", and "#".

6. The method of claim 3, wherein the at least one
wildcard comprises an absence of characters in a period
delimited sequence of domains.

25

7. The method of claim 3, wherein the at least one
wildcard comprises a wildcard for an entire domain label.

8. The method of claim 3, wherein the at least one
30 wildcard comprises a wildcard for a portion of a domain label
that is less than the entire domain label.

9. The method of claim 2, wherein the at least one
operator comprises at least one Boolean operator.

10. The method of claim 1, further comprising transmitting the one or more identified domain names to the remote network computer.

5

11. The method of claim 10, wherein transmitting the one or more identified domain names comprises constructing user interface instructions including the identified domain names.

10

12. The method of claim 11, wherein the user interface instructions comprise HTML (HyperText Markup Language) instructions.

15

13. The method of claim 10, wherein transmitting the identified domain names comprises transmitting a network address of a host storing the constructed user interface instructions.

20

14. The method of claim 1, further comprising transmitting a network address corresponding to at least one of the identified domain names to the remote computer.

25

15. The method of claim 14, further comprising determining if identifying yields only one matching domain name; and transmitting to the remote computer a network address corresponding to the one matching domain name.

30

16. The method of claim 1, wherein identifying comprises accessing a database storing domain names and corresponding network addresses.

17. The method of claim 16, wherein the network addresses comprise IP (Internet Protocol) addresses.

18. The method of claim 16, wherein the stored domain
5 names comprise domain names having at least one blank space separating two non-blank characters.

19. The method of claim 1, wherein receiving a query for a network domain name comprises receiving a DNS (domain
10 name service) query issued by a resolver.

20. The method of claim 1, further comprising providing user interface instructions for transmission to a remote computer and presentation to a user; and
15 wherein receiving the query comprises receiving a query generated in response to user interaction with the user interface instructions.

21. The method of claim 20,
20 wherein the user interface instructions comprise HTML instructions; and wherein user interaction comprises text entry into a user entry field presented by the user interface instructions.

25 22. A computer program product, disposed on a computer readable medium, for handling a network domain name query, the computer program product comprising instructions for causing a processor to:

30 receive a query for a network domain name from a remote network computer, the domain name query capable of matching multiple, different domain names; and

identify one or more network domain names matching the received network domain name.

23. The computer program of claim 22, wherein the domain name query comprises at least one operator.

5 24. The computer program of claim 23, wherein the at least one operator comprises at least one wildcard.

25. The computer program of claim 23, wherein the at least one operator comprises at least one Boolean operator.
10

26. The computer program of claim 22, further comprising transmitting the one or more identified domain names to the remote network computer.

15 27. The computer program of claim 26, wherein transmitting the one or more identified domain names comprises constructing user interface instructions including the identified domain names.

20 28. The computer program of claim 22, wherein identifying comprises accessing a database storing domain names and corresponding network addresses.

29. The computer program of claim 22, wherein receiving
25 a query for a network domain name comprises receiving a DNS (domain name service) query issued by a resolver.

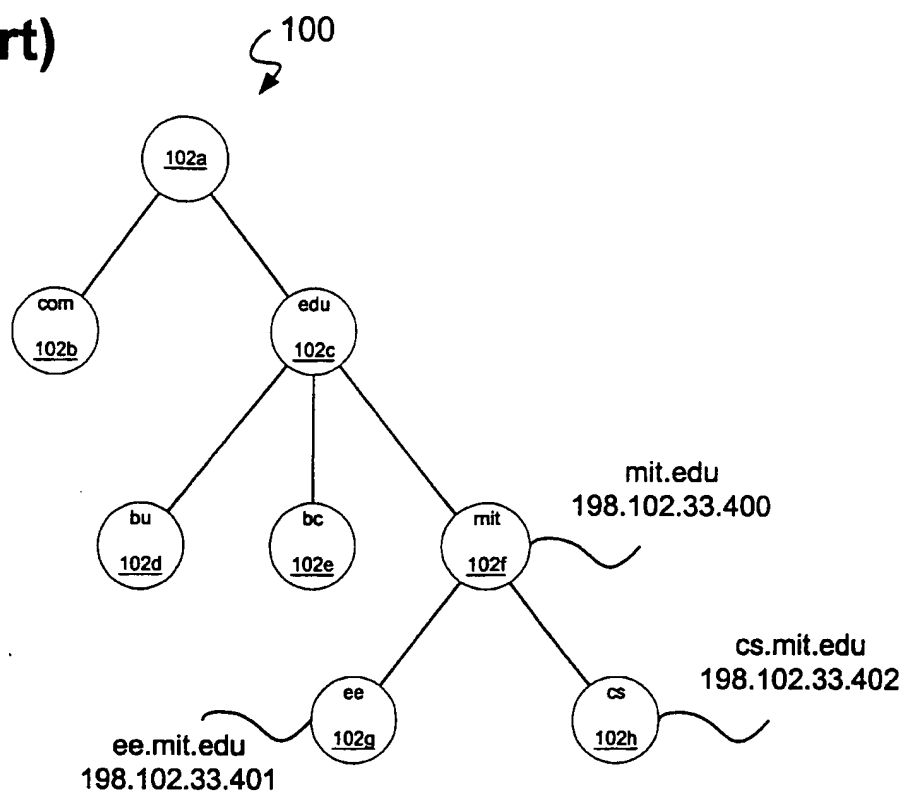
FIG. 1
(prior art)

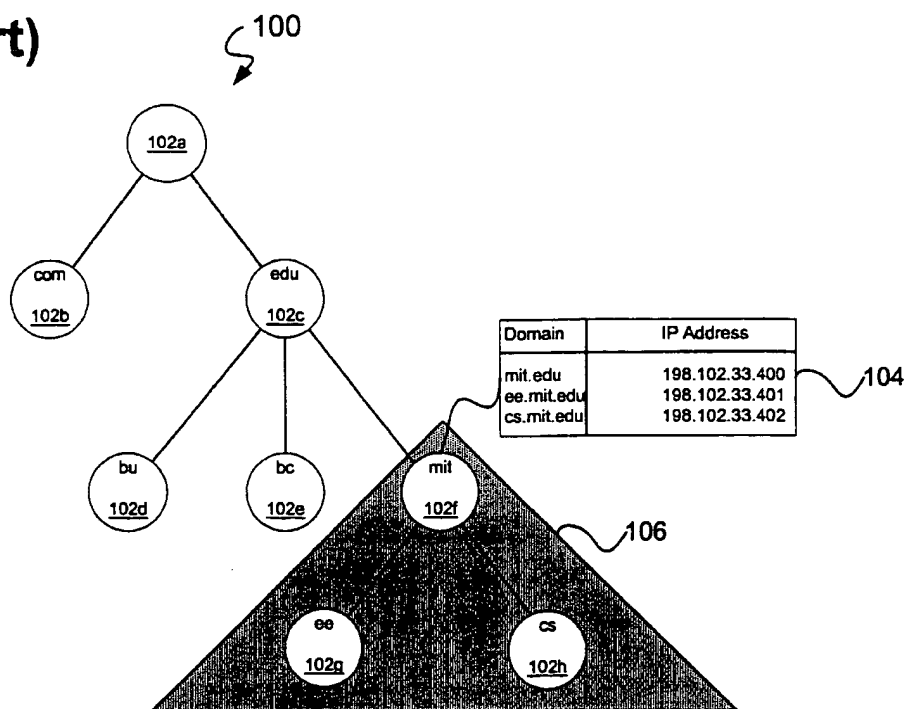
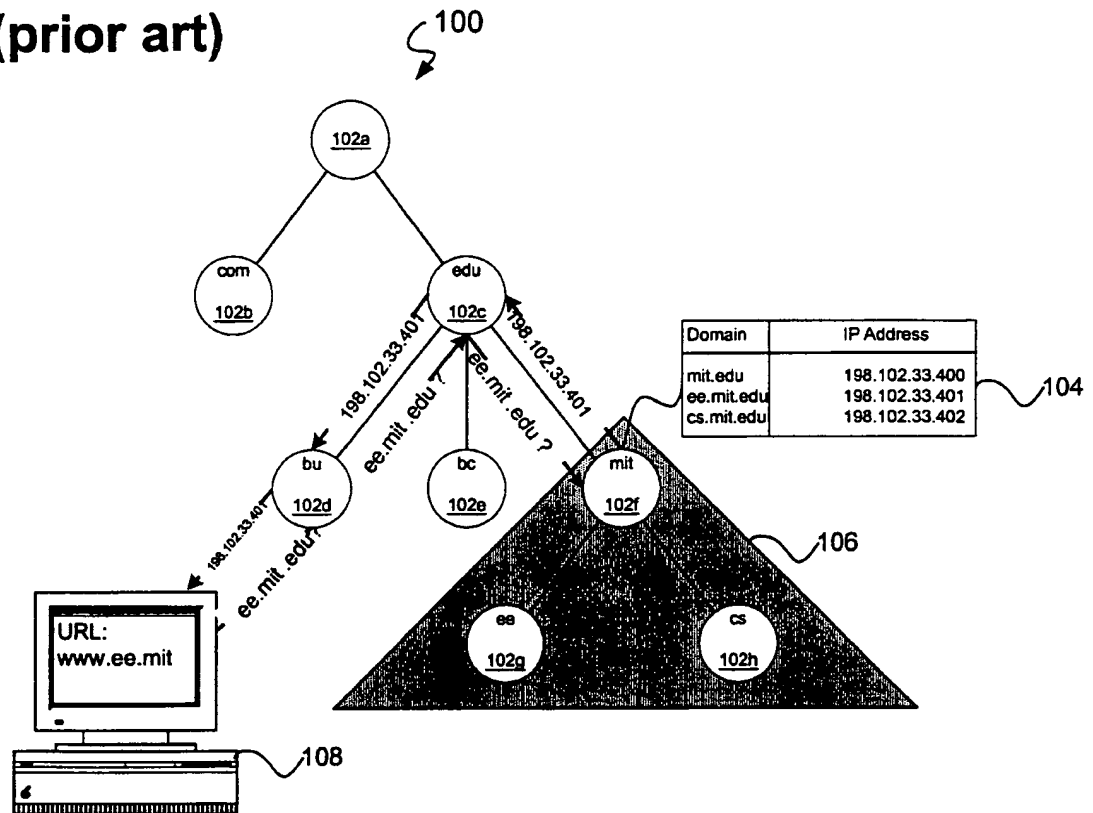
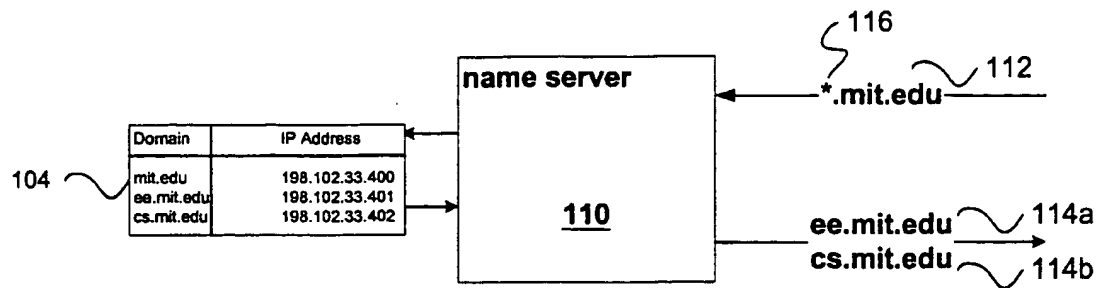
FIG. 2
(prior art)

FIG. 3
(prior art)

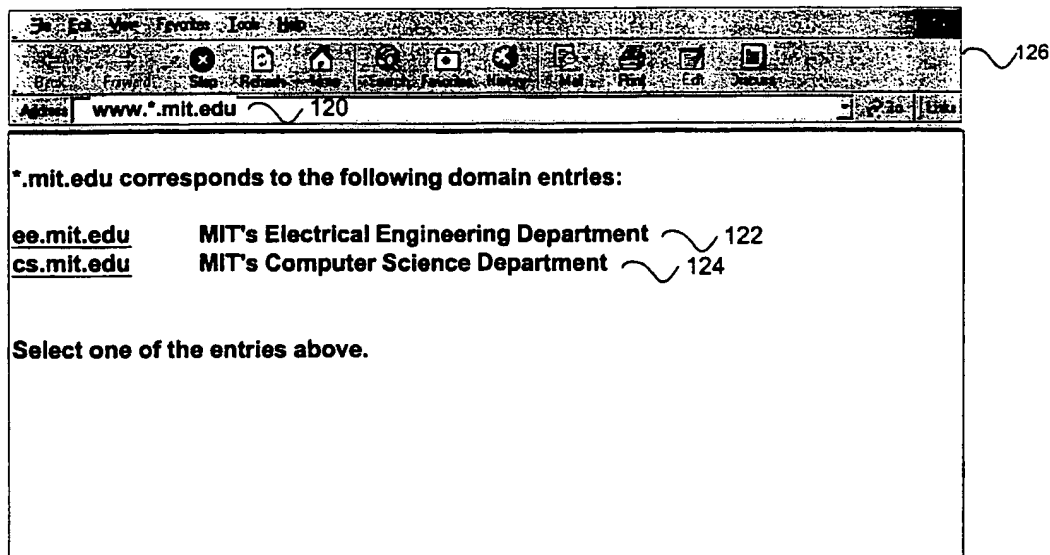
4/7

FIG. 4



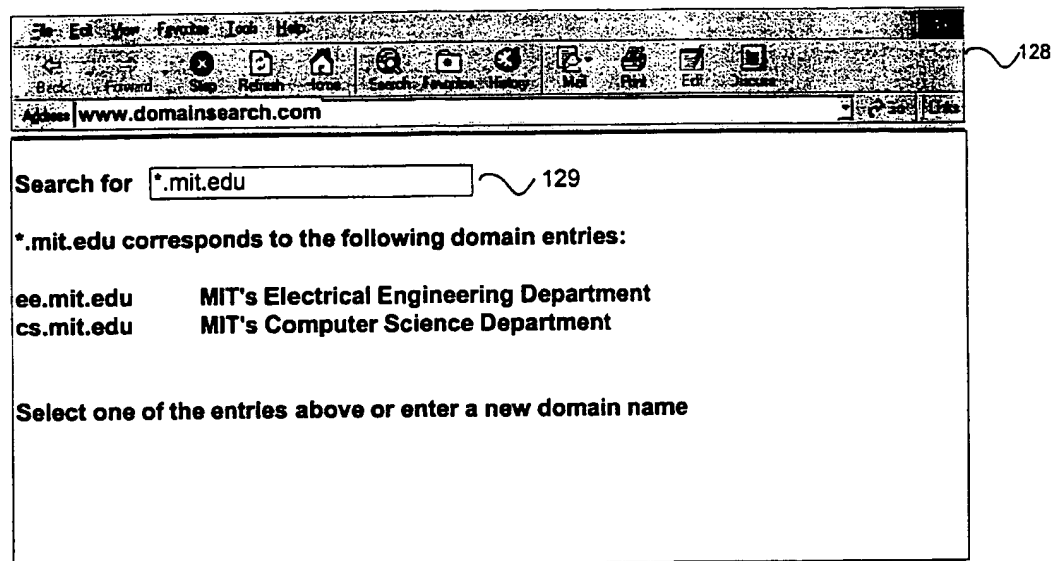
5/7

FIG. 5



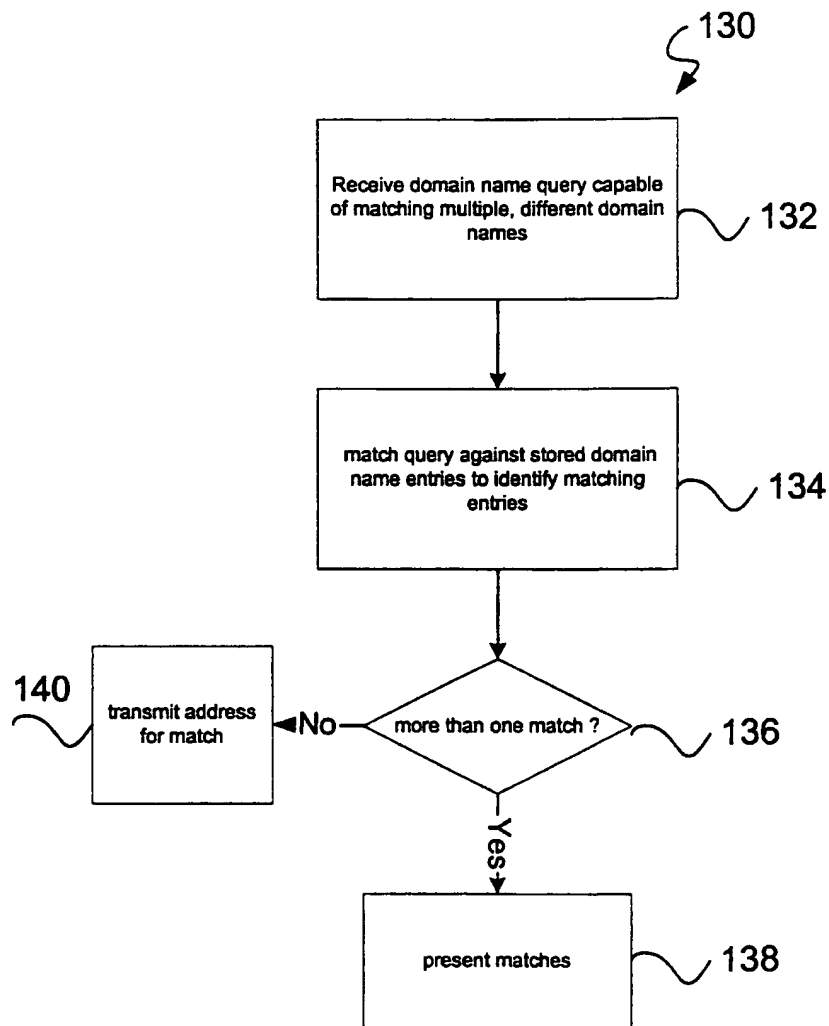
6/7

FIG. 6



7/7

FIG. 7



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.